Claims

What is claimed is:

1. A core product load manifest for protecting ongoing system integrity of a software product having a plurality of pieces; said core product load manifest comprising:

a manifest header including header attributes of the software product;

a list including a plurality of manifest items stored with said manifest header; each manifest item identifying a corresponding piece of the software product; each manifest item including at least one attribute; and

a manifest digital signature stored with said manifest header; said manifest header, said header attributes, each of said plurality of items, and each said item attribute included in said manifest digital signature.

2. A core product load manifest for protecting ongoing system integrity of a software product as recited in claim 1 wherein said at least one attribute of each manifest item includes a predefined attribute identifying the corresponding piece of the software product as signed or unsigned.

3. A core product load manifest for protecting ongoing system integrity of a software product as recited in claim 2 wherein a digital signature of said signed corresponding piece of the software product is stored with said signed corresponding piece of the software product and said signature is excluded from said manifest item identifying said signed corresponding piece of the software product.

4. A core product load manifest for protecting ongoing system integrity of a software product as recited in claim 3 wherein said signature of each signed corresponding piece of the software product and said manifest digital signature include a single certificate.

5. A core product load manifest for protecting ongoing system integrity of a software product as recited in claim 1 wherein an amended manifest is generated for identifying added and deleted pieces of the software product.

6. A core product load manifest for protecting ongoing system integrity of a software product as recited in claim 5 wherein said amended manifest is chained to the core product load manifest.

7. A method for protecting ongoing system integrity of a software product having a plurality of pieces; said method comprising the steps of:

creating a product load manifest for the software product; said product load manifest including a manifest header with header attributes of the software product; a list of a plurality of manifest items; each manifest item identifying a corresponding piece of the software product; each manifest item including at least one attribute identifying the corresponding piece of the software product as signed or unsigned; and a manifest digital signature stored with said manifest header; said manifest header, said header attributes, each of said plurality of items, and each said item attribute included in said manifest digital signature;

computing a digital signature of each signed piece of the software product; and

storing each said computed digital signature with the signed piece of the software product and excluding each said computed digital signature from said product load manifest.

8. A method for protecting ongoing system integrity of a software product as recited in claim 7 wherein the step of creating said product load manifest for the software product includes the step of receiving a certificate and copying said certificate into said manifest header.

9. A method for protecting ongoing system integrity of a software product as recited in claim 8 wherein the step of computing said digital signature of each signed piece of the software product includes the step of utilizing said certificate and a private key for computing said digital signature of each signed piece of the software product.

10. A method for protecting ongoing system integrity of a software product as recited in claim 7 further includes the step of creating an amended manifest for identifying added and deleted pieces of the software product.

11. A method for protecting ongoing system integrity of a software product as recited in claim 10 includes the step of chaining said amended manifest to said product load manifest.

12. A method for protecting ongoing system integrity of a software product as recited in claim 11 wherein the step of chaining said amended manifest to said product load manifest includes the step of including a pointer in said manifest header of said product load manifest to said amended manifest.

13. A method for protecting ongoing system integrity of a software product as recited in claim 11 includes the step of creating a next amended manifest for identifying added and deleted pieces of the software product; and generating a single linked list for chaining said next amended manifest and said amended manifest to said product load manifest.

14. A method for protecting ongoing system integrity of a software product as recited in claim 13 wherein the step of generating said single linked list for chaining said next amended manifest and said amended manifest to said product load manifest includes the step of providing a pointer in said manifest header of said product load manifest to said amended manifest and providing a pointer in a manifest header of said amended manifest to said next amended manifest.

15. A method for protecting ongoing system integrity of a software product as recited in claim 7 wherein the step of creating said product load manifest for the software product includes the step of including a pattern attribute in said manifest header attributes of the software product.

16. A computer program product for implementing ongoing system integrity protection of a software product having a plurality of pieces; said computer program product including a plurality of computer executable instructions stored on a computer readable medium, wherein said instructions, when executed by a computer, cause the computer to perform the steps of:

creating a product load manifest for the software product; said product load manifest including a manifest header; a list of a plurality of manifest items; each manifest item identifying the corresponding piece of the software product and including at least one attribute identifying the corresponding piece of the software product as signed or unsigned; and a manifest digital signature; said manifest header, each of said plurality of items, and each said item attribute included in said manifest digital signature;

computing a digital signature of each signed piece of the software product; and

storing each said computed digital signature with the signed piece of the software product and excluding each said computed digital signature from said product load manifest.

17. A computer program product for implementing ongoing system integrity protection of a software product as recited in claim 16 wherein said instructions, when executed by a computer, further cause the computer to perform the steps of creating an amended manifest for identifying added and deleted pieces of the software product.

18. A computer program product for implementing ongoing system integrity protection of a software product as recited in claim 16 wherein said instructions, when executed by a computer, further cause the computer to perform the steps of generating a single linked list for chaining said amended manifest to said product load manifest.

19. A computer program product for implementing ongoing system integrity protection of a software product as recited in claim 16 wherein the step of creating said product load manifest for the software product includes the step of receiving a certificate and copying said certificate into said manifest header.

20. A computer program product for implementing ongoing system integrity protection of a software product as recited in claim 19 wherein the step of computing said digital signature of each signed piece of the software product includes the step of utilizing said certificate and a private key for computing said digital signature of each signed piece of the software product.